AMENDMENTS TO THE CLAIMS

Please cancel claims 20-21 and 41 without prejudice. Kindly amend claims 1, 8, 13, 22-23, 32-33 and 43 as shown in the following listing of claims. Kindly add new claims 44-83 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (currently amended) An apparatus for generating early status flags to enable early execution of a conditional instruction in a pipeline microprocessor having architected status flags, the apparatus comprising:

   a storage element, for accumulating early status flags corresponding to the architected status flags; and

   logic, coupled to said storage element, configured to update said early status flags in said storage element in response to early results of instructions preceding the conditional instruction, wherein said logic invalidates said early status flags if at least one of said early results of said instructions that modify said early status flags is invalid, thereby enabling, if said early status flags are valid, execution of the conditional instruction based on said early status flags prior to the microprocessor updating the architected status flags in response to final results generated for said preceding instructions, wherein said logic is further configured to copy said architected status flags to said early status flags and validate said early status flags if the microprocessor pipeline is flushed.

2. (original) The apparatus of claim 1, further comprising:

   a first stage of the microprocessor pipeline, wherein said logic generates said early status flags;

   a second stage of the microprocessor pipeline, for updating the architected status flags;

   wherein said first stage is earlier in the microprocessor pipeline than said second stage.

3. (original) The apparatus of claim 1, wherein said early status flags comprise one or more x86 architecture EFLAGS register status flags.

4. (original) The apparatus of claim 1, wherein said early results comprise results of instructions comprising a subset of the instruction set supported by the microprocessor.

5. (original) The apparatus of claim 4, wherein said subset of instructions comprises instructions for performing simple arithmetic operations.

6. (original) The apparatus of claim 4, wherein said subset of instructions comprises instructions for performing simple shift operations.

7. (original) The apparatus of claim 4, wherein said subset of instructions comprises instructions for performing simple Boolean operations.

8. (currently amended) The apparatus of claim 4, wherein said subset of instructions comprises instructions ~~commonly~~ used for updating the architected status flags for use as condition codes specified by conditional branch instructions.

9. (original) The apparatus of claim 4, wherein said early results are generated prior to execution of the instructions by execution units of the microprocessor that generate final results of the instructions.

10. (original) The apparatus of claim 1, further comprising:

early execution logic, coupled to said logic, for generating said early results of said preceding instructions.

11. (original) The apparatus of claim 10, wherein said early results that modify said early status flags are valid if said instruction specifies an operation that said early execution logic is configured to perform and all input operands to said early execution logic used to generate said early results are valid.

12. (original) The apparatus of claim 10, wherein said early execution logic is within an address generation stage of the pipeline microprocessor.

13. (currently amended) The apparatus of claim 10, wherein said early execution logic generates said early results in response to source operands received from ~~is within a stage of the pipeline microprocessor immediately following a stage of the microprocessor including~~ an architected register file of the microprocessor.

14. (original) The apparatus of claim 10, further comprising:

an early register file, coupled to said early execution logic, having a plurality of registers corresponding to registers of an architected register file of the microprocessor, wherein said plurality of registers of said early register file are selectively valid.

15. (original) The apparatus of claim 14, wherein if one of said plurality of registers provides an input operand to said early execution logic to generate said early results of at least one of said instructions that modify said early status flags and said input operand is invalid, said logic invalidates said early status flags.

16. (original) The apparatus of claim 10, wherein said early execution logic is configured to execute a subset of instructions executable by the microprocessor, wherein if at least one of said instructions that modify said early status flags is not in said subset, said logic invalidates said early status flags.

17. (original) The apparatus of claim 1, wherein the pipeline microprocessor is a scalar microprocessor.

18. (original) The apparatus of claim 1, wherein the pipeline microprocessor issues instructions in program order.

19. (original) The apparatus of claim 1, wherein said storage element is not specifiable by program instructions.

20-21. (canceled)

22. (currently amended) A pipeline microprocessor having non-selectively valid architected status flags, and including in its instruction set conditional instructions that specify a condition and an operation, wherein if the condition is satisfied the microprocessor performs the operation, comprising:

early status flags corresponding to the status flags stored in the architected register, wherein said early status flags are selectively valid; ~~and~~

~~early~~ conditional execution logic, coupled to receive said early status flags, for performing an operation specified by a conditional instruction if said early status flags are valid, and if a condition specified by said conditional instruction is satisfied in said early status flags; and

control logic, coupled to said early status flags, configured to copy said architected status flags to said early status flags and validate said early status flags if the microprocessor pipeline is flushed.

23. (currently amended) The microprocessor of claim 22, further comprising:

final execution logic, coupled to receive the architected status flags, for performing said operation if said condition is satisfied in the architected status flags and said operation is not performed by said ~~early~~ conditional execution logic.

24. (original) The microprocessor of claim 22, further comprising:

logic, for generating said early status flags in response to an instruction preceding said conditional instruction, wherein said preceding instruction specifies modification of the architected status flags.

25. (original) The microprocessor of claim 24, further comprising:

an early register file, operatively coupled to said logic, having a plurality of registers corresponding to registers of an architected register file of the microprocessor, wherein said plurality of registers of said early register file are selectively valid.

26. (original) The microprocessor of claim 25, wherein said preceding instruction specifies at least one input operand from said architected register file, wherein said logic generates said early status flags in response to a result of said preceding instruction that is generated based on said at least one input operand provided by said early register file rather than by said architected register file.

27. (original) The microprocessor of claim 26, wherein if said at least one operand provided by said early register file is invalid, said early status flags are invalid.

28. (original) The microprocessor of claim 24, wherein if at least one input operand provided to said preceding instruction is invalid, said early status flags are invalid.

29. (original) The microprocessor of claim 24, further comprising:

early execution logic, coupled to said logic, configured to execute a subset of instructions executable by the microprocessor, wherein if said preceding instruction is not in said subset, said early status flags are invalid.

30. (original) The microprocessor of claim 22, wherein the microprocessor is a scalar microprocessor.

31. (original) The microprocessor of claim 22, wherein the microprocessor issues instructions in program order.

32. (currently amended) The microprocessor of claim 22, wherein said ~~early~~ conditional execution logic is within an address generation stage of the pipeline microprocessor.

33. (currently amended) A method for generating status flags early in a pipeline microprocessor to enable early execution of an instruction that conditionally performs an operation based on a condition of the status flags specified by the instruction, the method comprising:

generating a first instance of the status flags in response to an instruction preceding the conditional instruction wherein the first instance may not be valid;

generating a second instance of the status flags in response to the preceding instruction, subsequent to said generating the first instance, wherein the second instance is always valid; ~~and~~

performing the operation, prior to said generating the second instance, if the condition is satisfied in the first instance of the status flags and if the first instance is valid;

determining whether the microprocessor pipeline is flushed; and

copying architected status flags to the first instance of the status flags and validating the first instance of the status flags if the microprocessor pipeline is flushed.

34. (original) The method of claim 33, further comprising:

updating an architected register for storing the status flags after said generating said second instance.

35. (original) The method of claim 33, further comprising:

generating a result of the preceding instruction, prior to said generating the first instance;

determining whether the result of the preceding instruction is valid; and

invalidating the first instance if the result is invalid.

36. (original) The method of claim 35, wherein said determining whether the result of the preceding instruction is valid comprises:

determining whether the result of the preceding instruction is generated using valid input operands; and

indicating the result is invalid if the input operands are invalid.

37. (original) The method of claim 35, wherein said determining whether the result of the preceding instruction is valid comprises:

determining whether the preceding instruction is an instruction that is performable by early execution logic of the microprocessor; and

indicating the result is invalid if the preceding instruction is not an instruction that is performable by the early execution logic.

38. (original) The method of claim 37, wherein the early execution logic generates the result prior to generation of an always valid instance of the result by an execution unit of the microprocessor.

39. (original) The method of claim 35, further comprising:

determining whether the preceding instruction modifies the status flags; and

said invalidating the first instance only if the preceding instruction modifies the status flags.

40. (original) The method of claim 33, wherein said generating the first instance is performed without stalling the microprocessor pipeline regardless of whether input operands to the preceding instruction are valid.

41. (canceled)

42. (original) The method of claim 33, further comprising:

determining whether all status flag-modifying instructions present in the microprocessor pipeline below a stage in which said generating the first instance is performed, if any, have updated architected status flags of the microprocessor; and

copying architected status flags to the first instance of the status flags and marking the first instance of the status flags valid if all status flags-modifying instructions present in the microprocessor pipeline below a stage in which said generating the first instance is performed, if any, have updated architected status flags of the microprocessor.

43. (currently amended) A computer ~~data signal~~ program embodied ~~in~~ on a ~~transmission~~ computer-readable medium, comprising:

computer-readable program code for providing an apparatus for generating early status flags to enable early execution of a conditional instruction in a pipeline microprocessor having architected status flags, said program code comprising:

first program code for providing a storage element, for accumulating early status flags corresponding to the architected status flags; and

second program code for providing logic, coupled to said storage element, configured to update said early status flags in said storage element in response to early results of instructions preceding the conditional instruction, wherein said logic invalidates said early status flags if at least one of said early results of said instructions that modify said early status flags is invalid, thereby enabling, if said early status flags are valid, execution of the conditional instruction based on said early status flags prior to the microprocessor updating the architected status flags in response to final results generated for said preceding instructions, wherein said logic is further configured to copy said architected status flags to said early status flags and validate said early status flags if the microprocessor pipeline is flushed.

44. (new) An apparatus for generating early status flags to enable early execution of a conditional instruction in a pipeline microprocessor having architected status flags, the apparatus comprising:

a storage element, for accumulating early status flags corresponding to the architected status flags; and

logic, coupled to said storage element, configured to update said early status flags in said storage element in response to early results of instructions preceding the conditional instruction, wherein said logic invalidates said early status flags if at least one of said early results of said instructions that modify said early status flags is invalid, thereby enabling, if said early status flags are valid, execution of the conditional instruction based on said early status flags prior to the microprocessor updating the architected status flags in response to final results generated for said preceding instructions, wherein said logic is further configured to copy said architected status flags to said early status flags and validate said early status flags if all status flag-modifying instructions present in the microprocessor pipeline below a stage in which said early status flags are generated, if any, have updated architected status flags of the microprocessor.

45. (new) The apparatus of claim 44, further comprising:

a first stage of the microprocessor pipeline, wherein said logic generates said early status flags;

a second stage of the microprocessor pipeline, for updating the architected status flags;

wherein said first stage is earlier in the microprocessor pipeline than said second stage.

46. (new) The apparatus of claim 44, wherein said early status flags comprise one or more x86 architecture EFLAGS register status flags.

47. (new) The apparatus of claim 44, wherein said early results comprise results of instructions comprising a subset of the instruction set supported by the microprocessor.

48. (new) The apparatus of claim 47, wherein said subset of instructions comprises instructions for performing simple arithmetic operations.

49. (new) The apparatus of claim 47, wherein said subset of instructions comprises instructions for performing simple shift operations.

50. (new) The apparatus of claim 47, wherein said subset of instructions comprises instructions for performing simple Boolean operations.

51. (new) The apparatus of claim 47, wherein said subset of instructions comprises instructions used for updating the architected status flags for use as condition codes specified by conditional branch instructions.

52. (new) The apparatus of claim 47, wherein said early results are generated prior to execution of the instructions by execution units of the microprocessor that generate final results of the instructions.

53. (new) The apparatus of claim 44, further comprising:

early execution logic, coupled to said logic, for generating said early results of said preceding instructions.

54. (new) The apparatus of claim 53, wherein said early results that modify said early status flags are valid if said instruction specifies an operation that said early execution logic is configured to perform and all input operands to said early execution logic used to generate said early results are valid.

55. (new) The apparatus of claim 53, wherein said early execution logic is within an address generation stage of the pipeline microprocessor.

56. (new) The apparatus of claim 53, wherein said early execution logic generates said early results in response to source operands received from an architected register file of the microprocessor.

57. (new) The apparatus of claim 53, further comprising:

an early register file, coupled to said early execution logic, having a plurality of registers corresponding to registers of an architected register file of the microprocessor, wherein said plurality of registers of said early register file are selectively valid.

58. (new) The apparatus of claim 57, wherein if one of said plurality of registers provides an input operand to said early execution logic to generate said early results of at least one of said instructions that modify said early status flags and said input operand is invalid, said logic invalidates said early status flags.

59. (new) The apparatus of claim 53, wherein said early execution logic is configured to execute a subset of instructions executable by the microprocessor, wherein if at least one of said instructions that modify said early status flags is not in said subset, said logic invalidates said early status flags.

60. (new) The apparatus of claim 44, wherein the pipeline microprocessor is a scalar microprocessor.

61. (new) The apparatus of claim 44, wherein the pipeline microprocessor issues instructions in program order.

62. (new) The apparatus of claim 44, wherein said storage element is not specifiable by program instructions.

63. (new) A pipeline microprocessor having non-selectively valid architected status flags, and including in its instruction set conditional instructions that specify a condition and an operation, wherein if the condition is satisfied the microprocessor performs the operation, comprising:

early status flags corresponding to the status flags stored in the architected register, wherein said early status flags are selectively valid;

conditional execution logic, coupled to receive said early status flags, for performing an operation specified by a conditional instruction if said early status flags are valid, and if a condition specified by said conditional instruction is satisfied in said early status flags; and

control logic, coupled to said early status flags, configured to copy said architected status flags to said early status flags and validate said early status flags if all status flag-modifying instructions present in the microprocessor pipeline below a stage in which said early status flags are generated, if any, have updated architected status flags of the microprocessor.

64. (new) The microprocessor of claim 63, further comprising:

final execution logic, coupled to receive the architected status flags, for performing said operation if said condition is satisfied in the architected status flags and said operation is not performed by said conditional execution logic.

65. (new) The microprocessor of claim 63, further comprising:

logic, for generating said early status flags in response to an instruction preceding said conditional instruction, wherein said preceding instruction specifies modification of the architected status flags.

66. (new) The microprocessor of claim 65, further comprising:

an early register file, operatively coupled to said logic, having a plurality of registers corresponding to registers of an architected register file of the microprocessor, wherein said plurality of registers of said early register file are selectively valid.

67. (new) The microprocessor of claim 66, wherein said preceding instruction specifies at least one input operand from said architected register file, wherein said logic generates said early status flags in response to a result of said preceding instruction that is generated based on said at least one input operand provided by said early register file rather than by said architected register file.

68. (new) The microprocessor of claim 67, wherein if said at least one operand provided by said early register file is invalid, said early status flags are invalid.

69. (new) The microprocessor of claim 65, wherein if at least one input operand provided to said preceding instruction is invalid, said early status flags are invalid.

70. (new) The microprocessor of claim 65, further comprising:

early execution logic, coupled to said logic, configured to execute a subset of instructions executable by the microprocessor, wherein if said preceding instruction is not in said subset, said early status flags are invalid.

71. (new) The microprocessor of claim 63, wherein the microprocessor is a scalar microprocessor.

72. (new) The microprocessor of claim 63, wherein the microprocessor issues instructions in program order.

73. (new) The microprocessor of claim 63, wherein said conditional execution logic is within an address generation stage of the pipeline microprocessor.

74. (new) A method for generating status flags early in a pipeline microprocessor to enable early execution of an instruction that conditionally performs an operation based on a condition of the status flags specified by the instruction, the method comprising:

generating a first instance of the status flags in response to an instruction preceding the conditional instruction wherein the first instance may not be valid;

generating a second instance of the status flags in response to the preceding instruction, subsequent to said generating the first instance, wherein the second instance is always valid;

performing the operation, prior to said generating the second instance, if the condition is satisfied in the first instance of the status flags and if the first instance is valid;

determining whether all status flag-modifying instructions present in the microprocessor pipeline below a stage in which said generating the first instance is performed, if any, have updated architected status flags of the microprocessor; and

copying architected status flags to the first instance of the status flags and marking the first instance of the status flags valid if all status flags-modifying instructions present in the microprocessor pipeline below a

stage in which said generating the first instance is performed, if any, have updated architected status flags of the microprocessor.

75. (new) The method of claim 74, further comprising:

updating an architected register for storing the status flags after said generating said second instance.

76. (new) The method of claim 74, further comprising:

generating a result of the preceding instruction, prior to said generating the first instance;

determining whether the result of the preceding instruction is valid; and

invalidating the first instance if the result is invalid.

77. (new) The method of claim 76, wherein said determining whether the result of the preceding instruction is valid comprises:

determining whether the result of the preceding instruction is generated using valid input operands; and

indicating the result is invalid if the input operands are invalid.

78. (new) The method of claim 76, wherein said determining whether the result of the preceding instruction is valid comprises:

determining whether the preceding instruction is an instruction that is performable by early execution logic of the microprocessor; and

indicating the result is invalid if the preceding instruction is not an instruction that is performable by the early execution logic.

79. (new) The method of claim 78, wherein the early execution logic generates the result prior to generation of an always valid instance of the result by an execution unit of the microprocessor.

80. (new) The method of claim 76, further comprising:

determining whether the preceding instruction modifies the status flags; and

said invalidating the first instance only if the preceding instruction modifies the status flags.

81. (new) The method of claim 74, wherein said generating the first instance is performed without stalling the microprocessor pipeline regardless of whether input operands to the preceding instruction are valid.

82. (new) The method of claim 74, further comprising:

determining whether the microprocessor pipeline is flushed; and

copying architected status flags to the first instance of the status flags and validating the first instance of the status flags if the microprocessor pipeline is flushed.

83. (new) A computer program embodied on a computer-readable medium, comprising:

computer-readable program code for providing an apparatus for generating early status flags to enable early execution of a conditional instruction in a pipeline microprocessor having architected status flags, said program code comprising:

first program code for providing a storage element, for accumulating early status flags corresponding to the architected status flags; and

second program code for providing logic, coupled to said storage element, configured to update said early status flags in said storage element in response to early results of instructions preceding the conditional instruction, wherein said logic invalidates said early status flags if at least one of said early results of said instructions that modify said early status flags is invalid, thereby enabling, if said early status flags are valid, execution of the conditional instruction based on said early status flags prior to the microprocessor updating the architected status flags in response to final results generated for said preceding instructions, wherein said logic is further configured to copy said architected status flags to said early status flags and validate said early status flags if all status flag-modifying instructions present in the microprocessor pipeline below a stage in which said early status flags are generated, if any, have updated architected status flags of the microprocessor.